LEVEL II 12

TECHNICAL REPORT 80-04

Quarterly Technical Report:

The Design and Development [D2] of Generic Microcomputer-Based Command and Control [C2] Decision and Forecasting Systems

James F. Wittmeyer, III

CSM INC.

Computer Systems Management, Inc.

1300 WILSON BOULEVARD, SUITE 102 • ARLINGTON, VIRGINIA 22209

(12)

TECHNICAL REPORT 80-04

(9) Quarterly Technical Report. 1 Apr–3ϕ Jun 8ϕ

(6) # The Design and Development (D^2) of Generic Microcomputer-Based Command and Control (C^2) Decision and Forecasting Systems.

(14) CSM-TR-8ϕ-ϕ4

by

(10) James F. Wittmeyer, III

(11) 3ϕ Jun 8ϕ

(15) MDA9ϕ3-8ϕ-C-ϕ155, ARPA Order-3829

(12) 68

Computer Systems Management, Inc.
1300 WILSON BOULEVARD, SUITE 102
ARLINGTON, VIRGINIA 22209

411586

mt

TECHNICAL REPORT 80-04

12

QUARTERLY TECHNICAL REPORT:

THE DESIGN AND DEVELOPMENT (D^2) OF GENERIC MICROCOMPUTER-BASED COMMAND AND CONTROL (C^2) DECISION AND FORECASTING SYSTEMS

ARPA Order No.:	3829
Contractor:	Computer Systems Management, Inc. 1300 Wilson Boulevard Arlington, Virginia 22209
Effective Date of Contract:	11/5/79
Contract Expiration Date:	9/30/81
Contract No.:	MDA903-80-C-0155
Principle Investigator:	James F. Wittmeyer, III (703) 525-8585
Contract Period Covered:	4/1/80 - 6/30/80
Short Title of Work:	Generic Microcomputer-Based C^2 Systems

This research was sponsored by the Defense Advanced Research Projects Agency under ARPA Order Number 3829; Contract Number MDA903-80-C-0155; and Monitored by DSS-W. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either express or implied, of the Defense Advanced Research Projects Agency or the United States Government.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 80-04	2. GOVT ACCESSION NO.: AD-A088516
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Generic Microcomputer-Based C^2 Systems	5. TYPE OF REPORT & PERIOD COVERED: Quarterly Technical Report 4/1/80-6/30/80
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): James F. Wittmeyer, III	8. CONTRACT OR GRANT NUMBER(s): MDA903-80-C-0155
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Computer Systems Management, Inc. Suite 102, 1300 Wilson Boulevard Arlington, Virginia 22209	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Prder Number 3829
11. CONTROLLING OFFICE NAME AND ADDRESS: DARPA 1400 Wilson Boulevard Arlington, Virginia 22209	12. REPORT DATE: 6/30/80
	13. NUMBER OF PAGES: 61
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Defense Supply Service-Washington (DSS-W) The Pentagon Washington, D.C.	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Recommended for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Recommended for public release; distribution unlimited

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Command and control (C^2); computer-based systems; systems development; systems demonstration; microcomputers; information systems; decision systems; and forecasting systems

(C2) (C2) (D2)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The conduct of research into the nature and use of advanced command and control (C^2) computer-based decision and forecasting systems requires effort in the C^2 computer-based systems design, development, demonstration, transfer, and documentation areas. This report examines the design and development (D^2) of generic microcomputer-based systems via the identification and evaluation of DBMSs and statistical analytical systems/routines.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SUMMARY

This Quarterly Technical Report covers the period from April 1, 1980 to June 30, 1980. The Tasks/Objectives and/or Purposes of the overall project are connected with the design, development, demonstration and transfer of advanced command and control (C^2) computer-based systems; this report looks at the prospects for the development of generic microcomputer-based decision and forecasting systems. The Technical Problems thus addressed include the identification and evaluation of the components of a generic blueprint, including especially data base management systems and statistical analytical systems/routines. The General Methods employed involved classic literature survey and computer science performance evaluation techniques. Technical Results included the recommendation to modify the SEED and/or QDMS data base management systems for PDP 11 interface use with SPSS-11 and the adoption of a powerful 16-bit microcomputer (to replace current systems) on which new or vendor modified statistical analytical software can be written. Future Research will be conducted in the C^2 computer-based systems design, development, demonstration, and transfer areas.

CONTENTS

1.0 INTRODUCTION

The design, development (D^2) and application (transfer) of advanced command and control (C^2) computer-based decision and forecasting systems is the mission of the Defense Advanced Research Projects Agency's Defense Sciences Office's Cybernetics Technology Division's C^2 Decision and Forecasting Systems Program. Unfortunately, there are many problems connected with the design, development and transfer processes. This report examines many of these problems and suggests how a generic approach can alleviate many of the most serious problems and accelerate the cost-effective development and transfer of C^2 computer-based systems.

2.0 THE C^2 DECISION AND FORECASTING SYSTEMS PROGRAM

The Defense Advanced Research Projects Agency's Defense Sciences Office's Cybernetics Technology Division's (hereafter DARPA/CTD and CTD) Command and Control Decision and Forecasting Systems Program (C^2D&FS) has as its primary mission the design, development and application of advanced computer-based systems for enhanced C^2 processes especially as they involve the "commander" as a decision-maker and forecaster.[1] Since the basic and applied computer-based and non-computer-based research which underlies C^2 process enhancements is constantly evolving, it is necessary to survey and assess continually research problems and opportunities. Accordingly, this report presents the C^2D&FS Program goals (against the C^2 process), examines existing computer-based decision and forecasting systems developed under the auspices of the C^2D&FS Program, and presents an optimal generic design and development plan for improved computer-based systems production.

2.1 Requirements

Even though DARPA research programs are technically not born in direct response to Defense requirements, they must at least implicitly develop in connection with real requirements. Indeed, unlike in the 1960s and early 1970s, DARPA research programs must today pay much more than casual lip service to operational needs and problems.

Since the C^2D&FS substantive programmatic focus is upon the Defense Command and Control (or Command, Control and Communications [C^3]) process, it was and remains important to understand the process from an analytical perspective. Curiously, this critical process has seldom, if ever, been systematically studied out of context, that is, independent

of the <u>specific</u> requirements and functions of <u>particular</u> contexts, such as Naval C^2, tactical C^2, air and ground C^2, strategic C^2, tactical Naval C^2 and so forth. Consequently, there are no general analytical frameworks for the study of C^2.

Another problem has to do with the few specific "models" that do exist. All too frequently, these descriptions reduce to descriptions of the communications technology which in reality <u>underlie</u> the C^2 process. As Andriole has argued:

> "Definitions of 'communications, command, and control' (C^3) often characterize it as the ability to control weapons and maneuver units via sophisticated communications technology...while this perception encompasses a major and critical aspect of C^3, it does not encompass the full breadth of it. Communications, command, and control also includes the assimilation and analysis of C^3 information for use in decision making.
> However, the full utilization of today's sophisticated communications technology is largely dependent on the development of efficient information-processing methods. Thus, if communications technology continues to outpace advances in decision-making and decision-aiding methodology, one can reasonably expect C^3 problems to intensify."[2]

2.1.1 <u>The C^2 Process</u> - Andriole's concern for the <u>individual</u> in the C^2 process is shared by others who have recently begun to characterize C^2 as a set of decision-making, intelligence (hence, C^2I and C^3I), and forecasting functions affected by environmental, situational and perceptual factors and supported by the communications, computer, behavioral, and engineering sciences, as suggested by the following narrative typology constructed by Harris, Clarkson, and Fuller:[3]

- The cognitive functions of the "commander"
  - Perception of:
    - -- The internal well-being of the organization;
    - -- Threats to the organization;
    - -- Capabilities of the organization to act within the existing environment at each moment in time;
    - -- Response of the organization (both expected and actual) to direction given.
  - Decision-making in an environment bounded by:
    - -- Time constraints;
    - -- Traditional response patterns;
    - -- Historical analogues to current situations;
    - -- Organizational motives and goals;
    - -- Perception as set forth.
  - Direction-giving which is bounded by:
    - -- Limitations inherent in human communications;
    - -- Organizational reception capabilities and patterns;
    - -- Organizational capabilities at each point in time.
- Generic component elements of command and control
  - Inflow of information:
    - -- Statement of requirements for information;

--- To intelligence units;

--- To subordinate operational units;

--- To adjacent or cooperating operational units;

-- Information on own forces;

--- Status of subordinate combat and service elements;

--- Status of adjacent and cooperating units;

--- Status of potential reserves;

--- Reporting requirements--basic, as modified by combat/crisis situations;

-- Information on enemy;

--- From subordinate intelligence and operational units;

--- From intelligence units of higher headquarters;

--- Reporting on enemy capabilities, movement, location, communication security, ECM and radar capabilities;

--- Reporting requirements--basic, as modified by combat/crises;

--- Functions to be performed by total intelligence process at each command level, with sophistication and completeness dependent on size and capability of staff available.

- Staff functions in support of command and control

  - Operations:

-- Review incoming information - own and enemy forces;

-- Report on current status;

--- To commander;

--- To other staff elements;

--- By direction of commander, to higher headquarters, to adjacent/cooperating units;

-- Prepare new orders for subordinate units;

--- At direction of commander;

--- On own initiative;

-- Disseminate new orders on approval of commander;

- Planning:

-- Review incoming information - own and enemy forces;

-- Review current operations to establish base for planning future operations;

-- Prepare future plans for operations;

--- Direction of commander;

--- Own initiative;

-- Support operations staff in preparing orders for implementation of approved plans;

- Intelligence:

-- Review incoming intelligence information;

-- Collation;

-- Analysis/estimating of implications of new information;

-- Report preparation/briefings;

--- Commander;

--- Other staff elements;

--- By direction of commander, to higher headquarters and to adjacent/cooperating units;

--- Security process;

-- Based on requests from commander, other staff elements, and own initiave prepare requirements for information collection.

● Commander/decision-maker

- Supported by actions of staff and technical services:

-- On basis of his stated requirements (format, periodicity, detail of content, manner of presentation aids, etc.) and staff initiative, kept current on;

--- Intelligence of enemy;

--- Own force operations/capabilities;

--- Potential new operations/plans;

-- On own initiative, commander maintains personal communications with subordinate commander, adjacent commanders, higher headquarter commanders;

- Initiate activity by operations/planning staffs:

-- Prepare orders for change in current operations;

-- Plan for subsequent stages of operations;

- Initiate activity by intelligence staff:

-- Improve operations;

-- Gain new information;

- Issue orders for change in or new operations:

-- On basis of orders from higher headquarters on own initiative, but with approval of higher headquarters;

- Control/maintain oversight of response to his orders:

-- Set requirements for reporting;

-- Use of reconnaissance by own staff members;

● Technical support

- Communications - adequate functioning of communications network in combat environment. Network of facilities connecting subject command with higher and subordinate headquarters. Facilities must be:

-- Adequate to forseen information flow;

-- Secure;

-- Accurate in transmitting information;

-- Survivable/robust in combat environment forseen;

- Computer support:

-- Information handling; and

-- Decision aids.

2.1.2 <u>Computer-Based Leverage Points</u> - The above delineation of C^2 processes and functions clearly suggests the incredible complexity of the U.S. C^2 system. Yet, just as clearly we can see where computers can be employed productively.

First, it must be stated that in nearly all cases does contemporary mini- and microcomputer use require special purpose software. Accordingly, we are not suggesting here that today's computers are ready immediately for C^2 use, but rather that they are <u>capable</u> of same with some investment in software.

Mini- and microcomputers may thus be used in the following areas:

- Decision-making, via interactive decision analytic models;
- Forecasting, via interactive quantitative political, military, and economic systems;
- Training;
- Statistical analyses;
- Data storage and retrieval;
- Mini-simulations;
- (With compatible [multiple] systems) Routing and group decision-making;
- Management information systems use;
- Personnel agendizing and organization;
- Subjective (Bayesian) forecasting;
- Crisis management, via empirically based decision/information aids;
- Option screening and intelligence assessment;
- Reporting, via standardized reporting programs;
- Resource allocation;
- Record keeping;
- Equipment inventory;

- Maintenance scheduling;
- Readiness evaluation;
- Weapons check-out;
- Weapons operation simulation; and
- Mission planning, via interactive mapping techniques, and so forth.

2.2 Goals

The goals of the C^2D&FS Program descend from descriptions of the C^2 process and leverage point identifications similar to the ones presented above. However, unlike requirements-oriented C^2 research conducted in the Services and the Intelligence Community, DARPA C^2 research is by mission and definition necessarily "advanced" and experimental. It is consequently unique in the defense research community.

The specific goals and technical approach of the C^2D&FS Program appear below:

GOALS

- Facilitate fulfillment of C^2D&FS as an objective by developing/improving quantitative methodologies and computer-based technologies for decision-making and forecasting;
- Provide methodological and technological groundwork for enhancing performance in operations and I&W components of C^2 as well as facilitating their procedural linkage:
  - Methodological/Technological:
    - -- I&W: Develop/improve methodologies for forecasting, estimate generation, and assessment of soft, "non-quantifiable" data;

-- Operations: Develop and test technologies and methods for improved option generation and action selection;

- Procedural:

-- Understanding of the synergism of I&W and operations for improved C^2.

TECHNICAL APPROACH

- Based on the conception of C^2 as an objective and on intelligence and communications as means to:

  - Conduct basic research in decision-making:

    -- Decision Analysis

    -- Empirical outcome assessment;

    -- Psychological and artifical intelligence methods for analysis of adversary/ally choice and reaction;

  - Conduct basic research in forecasting:

    -- Develop/integrate new forecasting methodologies;

    -- Improve/integrate subjective and objective indicators;

    -- Basic research on assessment of intentions, perceptions, deceptions;

- Apply results from and test this basic research to/in areas in which I&W and/or operations components of C^2 are crucial, e.g.:

  - Military/political/economic crises;

  - Alliance C^2;

  - Counter-terrorism;

- Command psychophysiology; and
- Bargaining and negotiations.

2.2.1 <u>"Basic" C^2 Decision and Forecasting Research</u> - Implicit in the above presentation of the C^2D&FS Program's goals and technical approach is a commitment to the support of basic research, that is, research which will inform and contribute to the development of C^2 computer-based systems. To the extent that the Program's charter is broadly defined, no intellectual discipline is beyond its interest, including:

- Sociology;
- Organizational Theory;
- Communications Theory;
- Systems Theory;
- Military Science;
- Engineering;
- Psychology;
- Political Science;
- Physiology;
- Economics;
- History;
- Computer Science; and
- Diplomacy.

The basic C^2D&FS research which results from the support of individuals and organizations laboring in these and other disciplines must then be converted into C^2 applications.

2.2.2 Applied Computer-Based C^2 Decision and Forecasting Research - While one of the primary missions of the C^2D&FS Program is to develop and apply computer-based decision and forecasting systems, successful applications are not necessarily computer-based. It is our contention, however, that computer-based applications are the most useful and enduring. Accordingly, our emphasis from this point forward will be upon improving the design, development and transfer of C^2 computer-based decision and forecasting systems.

3.0 CHARACTERISTICS OF C^2 COMPUTER-BASED DECISION & FORECASTING SYSTEMS

In order to improve the design and development (D^2) of computer-based decision and forecasting systems it is necessary to identify and dissect their components. This will facilitate an understanding of where we are now in the D^2 process and how we might affect improvement.

3.1 Current Systems

Currently there are a variety of C^2 computer-based systems, including:

- The Early Warning and Monitoring System (EWAMS);
- The Executive Decision Aids;
- OPINT;
- EVAL;
- INFER;
- RAM;
- DECISION;
- The (Counter-) Terrorism Research and Analysis Program (TRAP);
- The Adaptive Information Selector (AIS);
- The Spatial Data (Base) Management Systems (SDMS);
- The Ultra-Rapid Reader; and
- The Marine Corps Combat Readiness Evaluation System (MCCRES), among others;
- PRESS

The above systems are essentially complete systems which have been transferred to at least some extent, and they are all applicable to C^2. Since they have enjoyed some success and experienced some failure, they can inform our efforts to improve the C^2 D^2 process. For example, what is it about the Bayesian decision aids which has yielded so much transfer success and the EWAMS which, after years of development, has yielded some difficulty in the transfer realm? What is the basis of SDMS and SDMS-like technology appeal? Why have the crisis management executive decision aids been rejected by potential "customers?" Why has the TRAP been so successful?

The first task is to describe these systems briefly in order to pinpoint similarities and differences for evaluation and D^2 purposes.

Accordingly, the Early Warning and Monitoring System (EWAMS) is an interactive computer-based system of international political indicators for the monitoring of the flows in international affairs and for the prediction of crisis and conflict between entities within the system. EWAMS currently includes quantitative numerical and descriptive (textual) international political data from 1966 to the present for all countries in the world. The sources for the data are the New York Times (NYT), Times of London (TOL) and Manchester Guardian (MAG) all encoded into World Event Interaction Survey (WEIS) format. The data provides the means to do the retrospective as well as current analysis.

The Executive Decision Aids allow an analyst to search data sets to determine the actions/objectives and historical precedents and analogies of post World War II crisis situations throughout the world. The XAIDS assist in identifying the I&W patterns that signal the onset of crisis and generate aids

to assist crisis managers after a crisis has begun relevant to the U.S., China, and the Soviet Union.

OPINT software provides computer-driven option screening and intelligence assessment. Using multi-variate decision techniques, an expected value matrix of option selection is generated. This is an aid to decision making when the key states variables are not known.

OPINT provides dyadic (two-factor) influence diagramming capability to aid decision makers to select from various related, uncertain options. The program includes tutorial information so it can be used by casual users.

The prototype version of this software aided decision makers in selecting the best posturing option for the 6th Fleet during the recent Lebanese evacuation crisis. It has also been used during various planning exercises throughout the European Command (EUCOM/J2, J3).

The EVAL software allows users to construct hierarchical decomposition evaluation models for the evaluation of complex systems. The user interactively provides the structure and labels, and assigns importance by means of weights. The system supports simultaneous comparison of up to five systems. Output of the system is the unit of merit (score) for each candidate being evaluated. Besides the final score, intermediate aggregation is displayable as well as discrimination at each level. A "roadmap" is produced which shows the key discriminators or factors which most significantly differentiate the contending systems.

Sensitivity Analysis is also provided to allow the user to determine the criticality of sets of importance weights. A

data base retrieval capability can be used to store descriptive summaries, making EVAL a useful briefing tool for higher level decision makers.

Prototype versions of this software have been used successfully in procurement cycles of the improved TOW Vehicle, shipboard intermediate range combat system, the single channel ground-to-air combat systems for the Department of Defense and for other system evaluations such as evaluation of the U.S. Military Academy.

INFER (HIER) is an inference modelling system which aids the user in building probability diagrams of hierarchical inference. These are most useful when the complexity of a real-world inference problem requires an amount or kind of knowledge beyond the capability of any one individual. In such cases, many different individuals with different expertise can decompose the problem along hierarchical lines, assessing those probabilities which link the data to intermediate variables to the main hypothesis.

RAM is a resource allocation system which enables users to perform quick systematic cost/benefit analyses. RAM has been used repeatedly for the Army and Marine Corps POMs (Program Objectives Memoranda).

The DECISION software allows users to interactively construct decision trees using four basic types of combinatorial rules: probability nodes, simple cumulative nodes, multiplicative nodes and decision nodes. These elements lead to a rather natural way to conceptualize and resolve complex decisions. The primary objective of DECISION is to model a decision, or some part of it, so that at least some of the implications can be deduced.

The Terrorism Research and Analysis Program (TRAP) software allows an analyst to investigate data representing known terrorist groups, their nomenclature, modus operandi, and the associations of individuals within terrorist organizations. TRAP is used for both data collection as well as retrieval and analysis. (Almost without exception the prototype of the TRAP software followed the development of the XAIDS. Thus everything stated about the XAID prototypes can also be said about TRAP, with one major exception: The TRAP software is not as inexpensively convertable to productive use on the DDF PDP 11/70 in FORTRAN IV Plus.)

The Adaptive Information Selector (AIS) is a computer program which simulates a user's selection, rejection and routing of intelligence messages in his absence or in association with him.

The Spatial Data (Base) Management System (SDMS) technology has been incarnated in a large screen display system, a microcomputer-based system and in a small screen, non-color minicomputer-based system. Succinctly, SDMS technology enables a (C^2) user to store, retrieve and process data spatially without the aid of a standard keyboard.

The Marine Corps Combat Readiness Evaluation System (MCCRES) is an EVAL-based readiness reporting and assessment system adopted by the Marines for on-line use.

3.2 Similarities

All of these systems have a lot in common. First, they are all mini- and/or microcomputer-based. They are all interactive. They all employ at least rudimentary graphic routines. They all process quantitative (subjective or objective) data. They all are self-prompting (save AIS). They are, to some

extent, all menu-driven. They all yield tabular displays, if desired. They are all relatively inflexible. They were all developed without explicit regard for operational requirements. They are all largely in a state of perpetual change.

3.3 Differences

Unfortunately, the above systems have a lot of dissimilarities as well. For example, they have been programmed in a variety of incompatible languages (APL, BASIC, FORTRAN, C). Only a few have a color display capability. Some are generic (Bayesian decision aids, SDMS, URR, AIS) while others are more substantively focused XAIDS, EWAMS, TRAP, MCCRES). (A glaring realization here is that applications and transfer successes are much more frequent when the C^2 system is generic.) Some process quantitative-empirical data while others quantitative-subjective. Some process video and audio data, some do not. Some use standard keyboard-based interactive sequences and others use non-standard techniques (such as joysticks, touch sensitive panels, voice input, and function keys). Some provide useful hard copy while others do not. Some are relatively easy to use (or invisible, like the AIS); others are extremely difficult to use even with (too) lengthy user's manuals. Some are reasonably user-oriented; others are barely so. Some have structured data base management systems and others have no data management systems at all.

3.4 Evaluation of Current Systems

Evaluation is a tricky business. Ultimately it depends upon the weights one assigns to system characteristics and performance. But it also depends upon how we choose to define characteristics and performance. Here additional problems are confronted: How does one define key characteristics and

performance?; can they be defined at all? Nevertheless, it is possible to offer some evaluative generalizations about the current systems assuming certain weights and definitions, as follows:

- If technology transfer (defined as actual operational use and substantial cost-sharing) is highly weighted, then the Bayesian decision aids are far and away the "best" systems yet developed and the crisis management executive decision aids the "worst";

- If technological innovation (defined simply as high inventive, that is, truly new, quality is heavily weighted then the SDMSs are the "best" and MCCRES the "worst"; and

- If the exploitation of basic research in the form of a computer-based systems is important then the URR and TRAP are the "best" and the Bayesian decision aids the "worst."

When we step back for a moment and think about these generalizations a number of insights come to mind. For example, why are interactive graphics so important when the most "transferred" systems (Bayesian decision aids) have virtually none? Relatedly, color output seems important only to the designer--not the intended user. Non-standard input devices also seem relatively unimportant when examined in the context of the Bayesian aids. Similarly, large empirical data bases by and large seem not to impress intended users. Instead, they worry them (because of "care and feeding" requirements). Systems that are relatively invisible to the user, such as the AIS and the heretofore undiscussed Logicon man-machine relations work, seem to be easier to transfer than those which interrupt normal procedure.

On the technological innovation side, systems which boast only advanced input/output devices and sequences generally fail to attract real users. (Indeed, one can argue that SDMS and URR are the only truly innovative existing C^2 systems from an input/output perspective.)

If we isolate the most successfully transferred systems, the Bayesian aids, we note that they are generic and therefore applicable to many classes of problems and therefore easily integrated into established operational routine. Users do not have to find a place for them. (A related success variable is the level of operational expertise which went into the development and transfer of the aids. Candidly, Decisions and Designs, Inc. [DDI] personnel, because of their first hand knowledge of real requirements, were able to accelerate transfer through informed system design and development. Conversely, the EWAMS is still a system in search of a solution, attempting to adapt to requirements after the fact, suggesting that [a] systems can and should not be retrofitted to user requirements and [b] C^2 systems development should follow a thorough requirements analysis.)

If we isolate unsuccessfully transferred systems, like the crisis management executive aids, we note that they appear to have been designed in a vacuum, informed only by intuitive judgements about how they might be used. This coupled with clumsy input/output/display technology and interactive sequences have made them difficult to transfer.

An interesting contrast is TRAP. Nearly as clumsy to operate as the XAIDS with difficult to interpret graphic output, it is a transfer success because of its perceived fulfillment of operational requirements (still, in reality, vague because of the newness of the counter-terrorism

operational imperative). This suggests that transfer success is a perfect function of perceived relevance to operational requirements. MCCRES is an obvious example of this phenomena.

Systems which must be perpetually updated and/or modified also seem to less successfully transferred than those that can be put in place and left unchanged. The realities of data updating (with concerns arising over cost, reliability, and credibility) frighten and bewilder users. Moreover, data updating requirements suggest--rightly or wrongly--additional work for users.

Finally, flexibility is critical to transfer success. But flexibility does not mean that a system have many capabilities. Instead, it means that a user can modify the system to his changing needs. The Bayesian decision aids, for example, programmed modularly in APL, permit on-line input/output/display modifications. This capability has proven invaluable on countless occasions and guards against system obsolescence.

4.0 THE DESIGN AND DEVELOPMENT (D^2) OF GENERIC MICROCOMPUTER-BASED C^2 DECISION AND FORECASTING SYSTEMS (C^2D&FSs)

The above evaluation suggests a number of guidelines for the D^2 of C^2D&FSs. Generic in this context thus assumes many characteristics. Certainly requirements should dominate the D^2 of new systems. This, however, is not to say that requirements should dominate exclusively. Rather, that if transfer is important (as it perennially is) then a requirements analysis should precede the D^2 of new systems. Advanced input/output/display devices and techniques and optimal hardware/software configurations should flow from real (not perceived) requirements. New systems should be flexible (as defined above) and applicable to classes and subclasses of substantive problem areas (like I&W in DIA, I&W in CIA, and so forth), easy to use, and easy to give away. Care and feeding requirements must be minimized. Finally, new systems should be interrelatable, not unlike the use of INFER and DECISION, for example.

But how? In the following sections some solutions are offered which, when taken together, suggest a new approach to the D^2 of C^2D&FSs.

4.1 Criteria

The D^2 of advanced computer-based D&FSs must be informed by D^2 criteria. Such criteria, when fruitfully applied, will enable us to D^2 systems of a new variety. The criteria and sub-criteria include:

- Requirements Analysis
  - Organizational/Bureaucratic;
  - Substantive;

- **Hardware**
  - Mainframe;
    - -- Minicomputer;
    - -- Microcomputer;
  - Storage Devices;
    - -- Hard;
    - -- Soft (Expandable);
  - Input Devices;
    - -- Keyboards;
    - -- Lightpen (gun);
    - -- Joystick;
    - -- Trackball;
    - -- Mouse;
    - -- Graphical Input Tablet;
    - -- Touch Panel;
    - -- Knee Control;
    - -- Speech;
  - Display Devices;
    - -- Refreshed CRT;
    - -- Storage Tube CRT;
    - -- Plasma Panel Display;
    - -- Teletypewriter;
    - -- Line Printer;
    - -- Laser Display;
    - -- Large-Screen Display; and
    - -- Graphical Display;

- Portability;
- Reliability;
- Appearance;
- Processing Speed;

- Software
  - Language;
  - Structure;
  - DBMS;
  - Statistical Packages/Routines;
  - Display Properties (Alphanumeric Characters);
    - Font;
    - Size;
    - Case;
    - Spacing;
    - Aspect Ratio;
    - Cursor;
  - Display Coding Techniques;
    - Shape Coding;
    - Color Coding;
    - Blink Coding;
    - Motion;
    - Depth;
    - Line Type;
    - Focus or Distortion;

- Interaction Mode/Dialogue Types

- Q&A;
- Form-filling;
- Menu-selection;
- Function Keys with Command Language;
- User-initiated Command Language;
- Query Language;
- Natural Language; and
- Interactive Graphics.

This long list suggests variables which are probably infrequently examined when a C^2 D&FS is designed and developed. Realistically, hardware, software, interactive dialogue types, and requirements are determined on the basis of what is already known and familiar to the developer (contractor). Seldom are systematic analyses conducted, and just as infrequently mis-targetted "masterpieces" are constructed.

4.1.1 <u>Requirements</u> - If the ultimate research objective is to apply technology then a requirements analysis should precede the D^2 process. Some questionnaire and survey methods of requirements analysis appear below:[4]

- Use of questionnaires to obtain ratings of the relative importance of various categories of information and system features;
  - Inexpensive. Difficult to be specific enough for detailed design decisions. Requires prior knowledge of all relevant information categories, although Delphi techniques might avoid this requirement;
- Use of questionnaires to obtain estimates of time spent on each task associated with recipient's job;

- Self-estimates of time spent on work activities are notoriously poor. If only relative time is required, this may be adequate;

- "Repertory Grid Technique", a questionnaire-based technique for determining user's "cognitive frame of reference";

  - Difficult to use successfully. High-level, and may not easily be made specific enough for detailed design decisions. Might be more useful for "personalized" systems than for capturing requirements of broad user class;

- "Delphi Technique", a survey technique in which recipient's responses are fed back, anonymously. Recipient responds again, while aware of previous responses of entire group;

  - Relatively expensive. Promotes consensus and identification of all information categories, but may suppress important individual differences. No instances found of application in specific area of computer systems design;

- "Policy Capture", one of several techniques for developing quantitative relationships between perceived system desirability and specific system features. In this case, relationship takes the form of a multiple-regression equation;

  - Somewhat expensive. Mathematical assumptions may be inappropriate. Paired-comparison procedure limits dimensionality. No instances found of application in specific area of computer systems design;

- Interviews with users to determine information requirements, decision points, organizational constraints, etc.

- Used more or less universally. Formal discussion in literature is mostly in the context of management information systems. Has many variants, such as "structured" interviews. Although a skilled interviewer can overcome some of the limitations of subjectivity and inability of users to verbalize their practices, these limitations are still signigicant. To apply this method at the detailed system design level requires an insightful user, or interviewer, or both. Most useful for preliminary data collection;

- "Ad Hoc Working Group", in which subject-matter experts devise system requirements by analysis and negotiation;

  - Appears somewhat effective at very high (undetailed) level. Has problems of subjectivity, and is susceptible to bias due to interpersonal relationships of group members (e.g., undue influence of high-status member). Probably irrelevant at detailed system design level;

- "Critical Incident Technique", in which users are asked, via interview or survey, for information about incidents of particular success or failure in the process of which the computer system will be a part;

  - A broadly useful technique which often yields significant insights into critical functions and information. Often used by human factors personnel, but not evident in the computer systems design literature;

- Job analysis techniques, such as task analysis, link analysis, and activity analysis, which attempt to characterize user behavior on the basis of direct observation;

  - Readily applicable to manual and clerical tasks, in which direct observation yields necessary raw data. Much more difficult to apply to congitive tasks. Nonetheless,

these techniques are broadly useful. "Task analysis" is often employed informally with inadequate detail and without necessary training in the technique.

Regardless of which method is selected the objective should be to zero in upon the administrative/bureaucratic and substantive target environment with specific regard to the kind of user likely to actually use the system as suggested below:[5]

- Naive users (inexperienced with computers)
  - Computer-naive users are actually a very heterogeneous group, but have many common properties. Naive users benefit greatly from computer-initiated dialogue, usually require more tutorial features. Correct implicit "mental model" of computer systems and interactive dialogue cannot be assumed, must be explicitly conveyed by system. Naive user population has many detailed implications for dialogue design. Smooth transition from naive to experienced user is often difficult in current systems;
- Managers (including military commanders, etc.)
  - Managers tend to have highly variable information needs; current systems are often too rigidly constraining to satisfy those needs. Managers tend to place high negative value on own effort, have considerable discretion with respect to mode of system use or nonuse. Thus, very low "impedance" is required to capture manager as direct user. If dissatisfied, manager tends to resort to "distant use" (interposing operator between manager and system) or partial use;
- Scientific and Technical

- High proportion report dissatisfaction with available automated tools. These users often respond to such dissatisfaction by becoming personally involved in design or implementation of software tools, or by altering task to match available tools.

A thorough requirements analysis will prevent the D^2 of systems which exist as solutions in search of problems. It will suggest and inform the selection of input/output/display/interaction types and probably prevent the use of a dim storage tube in a bright room!

While it is impossible to taxonomize here the requirements of all possible transfer sites, we can characterize the usual target "user" as naive. As a result we should D^2 systems tailored to those who are by and large unfamiliar with interactive computer-based systems of the DARPA/CTD genre. This objective is sound because DARPA/CTD transfers over the past three years have been to naive users and, incidently, because, as Evans, Nickerson and Pew, Thompson, Eason, and Martin have pointed out, there are many known heterogeneous naive user characteristics which can inform D^2.

4.1.2 Hardware - The selection of a mainframe is the first critical decision which descends from the requirements analysis. Clearly, there are hundreds of mini- and microcomputers from which to choose or evaluate. Realistically, given our current minicomputer investment and microcomputer technological progress, our primary C^2 computer-based decision and forecasting system will be, on the minicomputer side, PDP 11/UNIX-based, and, in the future, microcomputer-based (perhaps supported by the PDP 11). Accordingly, the sections that follow will devote more attention to the optimal microcomputer configuration since, for all practical purposes, DARPA/CTD is locked into PDP-11/UNIX applications.

4.1.2.1 Microcomputer Mainframes - The C^2D&FS Program has a favorite microcomputer mainframe--the Tektronix 4051/4052. It has been used by several contractors for many applications, including CACI (MOTIVAID, Crisis Management Executive Aids, and TRAP), IPPRC (EWAMS, DEWAMS) and DDI (SURVAV). (Indeed, there have in reality been very few microcomputer applications supported by DARPA/CTD. The list includes the above, the Bayesian aids, the micro-SDMS, and the URR.)

There are hundreds of microcomputer systems on the marketplace today but little understanding about how to categorize and therefore select them. Three critical distinctions involve S-100, Non-S-100, and "Turnkey" microcomputer systems.

S-100 systems are those which utilize the MITS, Inc. 8080 microprocessor "bus" which was initially part of the Altair 8800. Competitors of the Altair all produced systems that used the same bus structure (the S-100) as the Altair and a standard was born. The S-100 in reality is nothing more than a special 100 line (wire) interconnection (devoid of circuitry) which permits the connection of many and varied peripherals to the CPU. The S-100 bus structure standard has stimulated the production of many imaginative memory and I/O device controller systems as well as more esoteric systems (such as speech synthesizers) enabling a CPU owner to "mix and match" memories, I/O systems, and other peripherals.

There are currently many S-100 system manufacturers as the figure below suggests:

Manufacturer	Products
Alpha Micro Systems	Offers a complete line of systems including a multiuser, multitasking, time-sharing disk operating system. The cpu used is a Western Digital WD-16, 16-bit microprocessor. Full line of peripherals including floppy and larger disks. Full support software including higher-level languages.
Byte, Inc.	Distributed through Byte Shops. Control panel or panelless system with 8080 cpu. Numerous interface and memory cards.
CGRS Microtech	Cpu is 6502. Kits, boards, or complete systems. Various hardware modules and floppy-disk capability. Support software includes disk operating system and a subset of BASIC.
Cromemco Inc.	Offers a full line of Z-80 systems with numerous peripherals including crt's, floppy disks, and color display modules. Full support software including 16K BASIC and FORTRAN.
Imsai Manufacturing Corp.	Full line of hardware based on the 8080 or 8048 (version of 8080). Various peripherals. Full support software includes FORTRAN IV, extended and commercial BASIC.
North Star Computers, Inc.	Floppy-disk-oriented Z-80 system with extended BASIC.
Pertec Computer Corp.	Pertec is the company that acquired MITS. Offers a full range of systems from personal computers to extensive business systems with applications software.
PolyMorphic Systems	A full line of hardware based on the 8080 including floppy-disk drives and crt displays. BASIC is 11K bytes. Support software includes disk operating system.
Processor Technology Corp.	Full line of hardware and software based on the 8080. Comprehensive disk operating system and software includes FOCAL (math language), FORTRAN, PILOT, and extended BASIC.
Realistic Controls Corp.	Offers the REX computer system based on the Z-80. Package includes keyboard, video interface, microfloppy, and 24K of RAM.
Vector Graphics Inc.	Full line of hardware based on the 8080. Text editing system using Diablo printer. Software includes floppy-disk operating system and BASIC interpreter.

There are also many S-100 "products," as suggested below:

Manufacturer	Products
Base 2 Inc.	Cpu's; memories.
Computalker Consultants	Speech synthesizers.
Cybercom, a Division of Solid State Music	Video interface.
DC Hayes Associates	Modems.
Dynabyte, Inc.	Memories; crt interface.
Electronic Memories & Magnetics Corp.	Memories.
Godbout Electronics	Memories; other boards.
Heuristics, Inc.	Speech recognition.
International Data Systems, Inc.	Modem; frequency counter; clock module.
Micropolis Corp.	Floppy-disk controllers and disks.
MiniTerm Associates, Inc.	Video interfaces.
Mountain Hardware, Inc.	Ac controller board.
National Multiplex Corp.	I/O boards.
Parasitic Engineering	Cpu's; memories.
Peripheral Vision, Inc.	Cassette; I/O boards.
Phonics, Inc.	Speech recognition.
S. D. Computer Products	Cpu's; memories.
Speech Technology Corp.	Voice generation.
Tarbell Electronics	Cassette and floppy-disk interfaces.
The Space Byte Corp.	Cpu module.
Thinker Toys™	Memories; bus cards; disk controllers; others.
Vandenberg Data Products	Memories.

Non-S-100 systems generally utilize the "motherboard" approach to interconnection. The motherboard generally contains the microprocessor chip, some RAM memory for the user, a system monitor in ROM or PROM, and I/O device controllers. To expand the Non-S-100 system one has to either add additional memory chips to the motherboard or add external plug-in modules.

Some Non-S-100 bus microcomputer systems utilize a semi-standard SS-50 bus and the GPIB (General Purpose Interface Bus) or IEEE bus, but these "standards" are much less standard than the S-100.

Some Non-S-100 bus systems are listed below:

Manufacturer	Products
Apple Computer, Inc.	Cpu is 6502. High-resolution color display interface, inherent keyboard, and limited other I/O. BASIC and other support software.
Commodore Business Machines, Inc.	Distributors of the PET and KIM-1. PET is an integrated hardware system with keyboard, display, and 6502 microprocessor. BASIC language.
Compucolor Corp. Intelligent Systems Corp.	Systems with built-in high-resolution color crt display and keyboard. Based on the 8080. Minifloppy diskette included. BASIC interpreter.
Heath Company	Microcomputer kits based on the 8080 or LSI-11 cpu (16-bit microprocessor compatible with Digital Equipment Corp. PDP-11). Crt displays, paper tape, and printers.
Midwest Scientific Instruments	Full line of 6800-based mirocomputer equipment compatible with SS-50 (SWTP) bus. Disk extended BASIC and disk operating system.
Ohio Scientific	Systems based on 6502 microprocessor. One system based on the 6502, Z-80, and 6800. Full line of support software including 8K BASIC and disk operating system.
Radio Shack, Division of Tandy Corp.	Z-80–based integrated computer system with keyboard, display, and extended BASIC. Cassette tape, floppy disks, printers, and other peripherals. Disk operating system and some applications software.
Southwest Technical Products Corp.	Microcomputer systems based on 6800 microprocessor with floppy disks, printers, cassettes, and other I/O devices. BASIC and disk operating system.
Technico, Inc.	System based on the 16-bit TMS 9900 microprocessor. Floppy or minifloppy, cassette, color video board, and other peripherals. BASIC and other support software.
The Digital Group	Wide range of systems and hardware including crt's, floppy disks, cassette drives, and printers. Many special-purpose hardware modules. Cassette tape and disk operating systems. Support software includes BASIC and OPUS.

Turnkey microcomputer systems generally consist of proprietary CPU design and a good deal of efficient software. The systems are also usually integrated in design and appearance, unlike S-100 and Non-S-100 systems.

Turnkey systems are generally selected because of their existing software and maintenance features. Yet, in the DARPA/CTD community, the software which comes with turnkey systems

such as the IBM 5110 and the Tektronix 4051/52 series are never used! Instead, new applications software is written, negating one of the chief advantages of a turnkey system purchase.

Nevertheless, some turnkey systems offer higher level programming language interpretors or compilers not offered in other systems. Moreover, since reliability and maintenance are often critically important to prospective users, turnkey systems offer advantages over other systems.

Some representative systems are listed below:

Manufacturer	System Characteristics
American Microcomputer Corp.	Z-80–based business system with crt, printer, dual floppy disk, full line of business software packages in OPUS. Minimum system: $10,000.
Applied Data Communications	Microcomputer with 48K RAM, 1K PROM, dual floppy disk, 60 cps keyboard teleprinter, desk, many other options including hard disk. Full business applications packages at $250 per package. Minimum system: $10,300.
Billings Computer Corp.	8080-based 48K system with dual floppy disk (480K), crt, 60 cps printer, communications capability, wide range of other I/O options. Full line of business software packages. Minimum system: $17,500.
Cado Systems Corp.	8080-based business system with dual floppy disk (1.23 million bytes), 150 cps printer, crt, other options. Full line of business applications software in compiler language. Minimum system: $14,000.
Computer Management Group, Inc.	Microcomputer system with 32K, dual diskettes (480K), crt, 120 cps printer, many other options. Full business applications packages. Minimum system: $9875.
IBM Corp.	Model 5110 computing system with 16K BASIC, keyboard, crt display, 120 cps printer, 2.4 megabyte diskette storage, other options. Typical system: $19,000. Accounting applications packages available.
Kerr & Reynolds	Pharmacy-oriented system with Cromemco Z-2 (Z-80), 32K, dual floppy disk, 108 cps printer, other options. Basic system: $18,000. Pharmacy software only.
Kitzmiller Systems	AM-100 microcomputer-based system with crt, 300 cpm printer, floppy disk storage of 500K. Full line of business software including "company workload management." About $16,000.
Logabax	8008-based office computer with keyboard, 180 cps printer, dual diskette storage. Invoicing, accounts receivable, other functions. About $15,000.

4.1.2.2 Input Devices - There are many, many kinds of input devices including keyboards, lightpens, joysticks, trackballs, mice graphical input tablets, touch panels, knee controls, and speech input devices. To a certain extent, if one selects a Non-S-100 bus microcomputer system or a turnkey system then the input medium is predetermined. S-100 systems offer somewhat more flexibility. After examining many systems, however, a number of selection guidelines emerge. First, a system should not have dual or multi-input options. This confuses a user and contributes to a bizarre sense of competition among the devices. Secondly, since most users are naive, standard A/N keyboards ought to dominate. This judgement is based upon the current performance of speech input systems and other input devices, such as spatial or touch panels, which have not yet been perfected into routine use. However, as soon as large vocabulary speech and spatial input systems are perfected--and provided that they can perform independently of required auxiliary systems--they should replace standard alpha-numeric input systems because they are inherently easier to use.

Interestingly, lightpens, according to survey research, are relatively unpopular input devices as are joysticks, track-balls, and mice; hence why use them?

4.1.2.3 Display Devices - As with the selection of input devices, display devices are to a great extent determined by one's selection of a mainframe. We know, however, that storage tube display devices can be bothersome, particularly in well lit environments. They also preclude the use of some programming techniques which rely inherently upon refresh characteristics. We also know that Plasma-type displays are often reliable.

On the other side we know that "hard copy" is a prerequisite to successful transfer and that noisy printers are horrible. Ideally, graphics are present in the display (only if they are prudently used) and graphics hard copy available.

Large screen display systems are an added capability which have proven exceedingly popular. Unfortunately, not all microcomputer systems generate the composite video output necessary to drive many large screen display systems. (The scan converter display is unacceptable.) Another problem with high resolution large screen displays is cost. Generally not a serious selection criteria in government research and development, cost can be prohibitive when a $60K investment may be required to display output from a $15K microcomputer system! Consequently, less expensive lower resolution systems are wise choices.

4.1.2.4 Portability - Portability is an interesting criterion. To some, portable means suitcase size; for others, it suggests crates. To us, portable means easily transportable via custom packaging. In our view, no matter how numerous the peripherals, they can be packaged in custom-made "crates" and sent across town, country or oceans. Nevertheless, we should avoid systems which are overly cumbersome or sloppily configured, i.e., systems that are abnormally large physically vis-a-vis their capabilities. (Realistically, most current systems are grouped similarly according to size and capabilities--there are no really relatively small powerful systems or large weak ones.)

4.1.2.5 Reliability - Reliability among the large vendors in terms of hardware quality is comparable; maintenance, however, is a very different matter. Maintenance costs and quality vary tremendously. Note, for example, our recent experiences with the maintenance provided by Standard Memory, DEC, Tektronix and IBM. Clearly IBM maintenance is superior to Standard Memory and Tektronix; DEC service is comparable but

expensive. When selecting a microcomputer configuration, then, maintenance should be considered.

4.1.2.6 Appearance - Appearance is in the eye of the beholder. How many times have we heard approvals for a (new) system's appearance on Monday and esthetic objections on Tuesday! Realistically, most naive computer users have no opinion; if the system actually helps them with their jobs, they tend to be happy. However, we have by and large failed to "cabinetize" out configurations attractively up to this point. By cabinetization we mean the organization of system components not unlike the manner in which sterophonic music components are organized in a single piece of furniture. This guards against rearranging offices and hastily erecting shelves when a system is introduced to a transfer environment. Indeed, it is curious how and why this all important dimension of appearance has been ignored.

4.1.2.7 Processing Speed - A popular misconception about microprocessors is that some are much faster than others. In truth, microprocessors are similar in speed but differ widely in overall throughput as a function of I/O devices and, most importantly, software configurations. Accordingly, as subsequent sections will illustrate, the real leverage against speed is with software.

4.1.3 Software - Predictably, software lies at the heart of mini- and microcomputer system performance--not hardware. Software is generally ignored, however, when D^2 a system. While we have been quick to compare the Tektronix 4051/52 with the IBM 5100/10, we have not looked seriously at the software strengths and weaknesses of these systems. (Nor have we paid enough attention to memory and peripheral considerations.)

4.1.3.1 Software Languages - Languages range from machine (CPU) language through assembly language and to the higher-level languages such as FORTRAN, BASIC, PASCAL and APL. Minicomputers offer a variety of languages in either interpretative or compiler form; microcomputers offer far less languages. There are, in turn, many varieties of each language; there is no standard whatsoever.

There are a variety of software systems which run on a mini- or microcomputer system: systems software, support software and applications software to name the most common.

There is no reason to alter the present process by which C^2D&FS software is generated on the GFE minicomputers. FORTRAN IV PLUS and "C" appear to have become the standard under the UNIX operating system. DARPA/CTD has thus decided to transfer PDP 11 software to other UNIX CULC FORTRAN IV PLUS, Tektronix PLOT 10 systems or transfer UNIX, CULC, and Tektronix PLOT 10 with the applications software (alternatively, DARPA/CTD can support the rewriting of the software to suit the transferee's requirements).

On the microcomputer side, however, the situation is very different. Since the transfer almost always involves transfer of the hardware itself, the selection of a language is less important. On the other hand, since microcomputers are inherently less powerful than minicomputers, software selection becomes important for performance reasons. For example, most microcomputer systems support BASIC. Why? Not because of efficiency. Rather--and as vendor literature explicitly states--because it is a beginner's language easily learned by novice business users. In addition, BASIC programs are virtually non-transferable from machine to machine (because of odd "PEEK" and "POKE" commands used to interface to the

hardware and operating system, among other problems).

Another problem involves the use of BASIC (and other languages) in an interpretive form rather than as a compiler. The result is slow(er) execution. (In fairness, it must be noted that an interactive interpreter can aid flexibility and program development.)

The figure below looks at some languages and compares their attributes.

Type of Language	Execution Speed	Average Development Time (Instructions/Day)	Number of Machine-Language Instructions Generated
Machine Language	Fastest	5	One for one
Assembly Language	Fastest	20	One or more for one
FORTRAN Compiler	Fast	30	Many
COBOL Compiler	Fast	30	Many
BASIC Compiler	Fast	40	Many
BASIC Interpreter	Slow	50	Most

The next figure looks at some processing response times across assembly-languages, compiler languages, and interpreter languages.

Function	Assembly-Language System	Compiler-Language System	Interpreter-Language System
Multiply 1000 numbers of various sizes.	1 ms	6 ms	6 s
Divide 1000 numbers of various sizes.	1.5 ms	9 ms	9 s
Insert a 20-character string in the middle of 1000 characters of text.	7.5 ms	75 ms	10 s
Sort (alphabetize) a list of 100 20-character names.	0.1 s	2 s	8 min
Merge 20 names into a list of 100 20-character names.	25 ms	0.5 s	2 min

The final chart presents some application operating times:

Application	Assembly-Language System	Compiler-Language System	Interpreter-Language System
Sort and print 1000 names for mailing list; 100 characters/entry; disk system.	25 min	25 min	105 min
Generate inventory report of 1000 items; 100 characters/item; disk system.	25 min	30 min	41 min
Response time for locating and display of one random account from 2000; disk system.	5 s	5 s	30 s

All of this suggests, at the most general level, that:

- if there is not much pure processing (without I/O) to be performed, then the inefficiency of a high-level language may be acceptable;
- if there is a great deal of processing, then a higher-level language might be unacceptable; and
- if higher-level language is called for in the light of acceptable response and applications times, then a higher-level compiler is preferable to an interpreter if speed is important.

When we turn to memory requirements, we note that the assembly-language system requires less than compiler- or interpreter-language systems, as the next figure suggests:

Language	Application Program—Number of Bytes of Storage Required
Assembly-Language System	4000
Compiler-Language System	8000
Interpreter-Language System	20,000

When we drop to a discussion of specific languages we have, really, few to choose from since microcomputer vendors support but a few--but the number is growing. We will, for example, see more and more manufacturers moving away from BASIC toward more powerful languages, such as APL, FORTRAN and, especially, PASCAL. Already compilers (not interpreters) for these languages exist with more coming all of the time. So which language should be the standard C^2D&FS language? Should there be a standard?

Since microcomputer storage (of all kinds) will continue to be an issue in the next decade, language statement conversion tables should guide our selection. For example, we know that FORTRAN is more powerful (requires less statements) than BASIC and that APL and PASCAL are more powerful than FORTRAN. We generally have not deviated too far from BASIC for two reasons: (1) our (Tektronix) microcomputers do not support other than BASIC and (2) the contractor's have limited expertise beyond BASIC (indeed, only a few have used APL and many have not even heard of PASCAL). The expertise problem is solvable; machine limits are too but we have to carefully consider all machine features before selecting one.

4.1.3.2 Data Base Management (DBM) Systems (DBMS) - Obviously there are many DBMS on the market today. Note that:

- Very few support DEC PDP 11 mainframes;
- Of those that support PDP 11 mainframes most do not run under UNIX (but conceivably could with reprogramming);
- There are very few commercially supplied micro DBMSs; and
- Current DBMS differ considerably in their structures and capabilities.

Note also that arrows highlight those realistically relevant to the generic task at hand.

However, it is important to understand what DBMS are and what they are not. Generally an acceptable DBMS incorporates the following facilities:

- Application program independence from the DBMS control programs;
- Support of the programming language(s) used in the corporate environment prior to installation;
- Utility programs to facilitate creation and maintenance of the data base(s);
- Facilities for data reorganization;
- The ability to effect data security and access limitation;
- Automatic restart capabilities in case of system failure, or the ability to recover operations manually with minimum effort; and
- System facilities for "fine tuning" of the DBMS.

In addition to the above, one can also look for transaction processing capabilities, an inquiry/response facility, and a report generator of some sort. Non-programmer utilization features, such as an English-language query facility, are rapidly gaining popularity. This type of capability may be provided either through an interface between the DBMS and an independent package or through a facility offered directly by the vendor of the DBMS.

Regarding implementation, most of the DBMSs can be installed to run under almost any operating system offered with a given manufacturer's computers. The DBMS is typically designed to be independent of the operating system--but, in fact, most systems are sensitive to logic changes in any operating system version.

Distinctions among DBMSs and Data Management Systems (DMSs) are also critical.

A data base management system can be defined as a software system intended to manage and maintain data in a non-redundant structure for the purpose of being processed by multiple applications. A data base management system organizes data elements in some predefined structure, and retains relationships between different data elements within the data base.

A data management system, on the other hand, is one that is intended primarily to permit access to, and retrieval from, already existing files, usually for a single application. Although a data management system may provide the capability to minimize data redundancy and centralize the storage of data, the principal intent of the system is to perform such functions as information retrieval, report generation, and inquiry for a single application. Informatics, Inc.'s MARK IV is an example

of the type of product included in this category. It has many of the qualifications found in a true data base management system (it does in fact, have its own file structures), except for the fact that MARK IV's primary intended use is for the processing of single application files.

Some products that started out as data management systems have grown with the times to the point where they have become full-blown data base management systems as well as data management systems. Examples are Infodata Systems' INQUIRE and Mathematica Products Group's RAMIS II. Through a series of enhancements and restructurings, each of these systems has grown into a bona-fide DBMS.

The DBMS/DMS distinction begs an important issue relevant to the D^2 of generic C^2D&FSs. Do we really need a full blown DBMS or something less powerful?

On the microcomputer side, we must realize that micro-DBMSs currently have few of the capabilities of real macro- or even minicomputer DBMSs. They do, however, manage data via file structures and permit fairly flexible sorting and retrieval. They are also interfaceable with applications programs.

4.1.3.3 <u>Statistical Packages/Routines</u> - There are a variety of statistical packages available. Obviously the most sophisticated run on macrosystems. We see a degredation of capability as we move down the computer system hierarchy until we reach the micro level where statistical packages consist of disjointed routines.

APPENDIX C presents some of the available packages/routines for conducting various statistical operations. The newest version of SPSS (#11) designed to operate under UNIX (among

other operating systems) is still the most promising. DYNASTAT is also very acceptable as is the PLATO and BMDP systems.

There are also statistical routines available for microcomputers (all of the above are mini- or macrocomputer system based). Some of these include:

- STATISTICS, a three-disk series is, according to Compucolor Corp., especially useful for engineering applications. Each disk contains five seperate microcomputer programs stored on a soft disk and comes with complete documentation;

- STATPAK, from Northwest Analytical, a statistical software library designed for the microcomputer user and small microcomputer system;

- Ohio Scientific Inc.'s OS-DMS QUOTATION/ ESTIMATION SYSTEM designed for non-computer oriented users. Software runs on Ohio Scientific Challenger II and III microcomputers;

- STATDIS, a turnkey microcomputer system for statistical analysis and display, comes complete with all equipment and software to provide data management, statistical analysis, graphic display in eight colors and hard copy printouts. Statdis equipment includes an 8-bit microcomputer with 24Kbytes of user memory, floppy disk storage of up to 2.4Mbytes, an 8-color crt with graphic display capability and a 100-cps dot matrix printer. The system, including all hardware and software, starts at $18,500. (Simcon Inc., Applied Microsystems Div., 7655 Old Springhouse Rd., McLean, VA 22102.); and

- The BUSINESS PLANNING PACKAGE for the TRS-80 is a floppy disk package containing a set of forecasting programs that will allow the user to solve a variety of

forecasting needs. The data preparation program allows the creation, modification, and deletion of disk based data sets. The data sets are accessible by all programs. (Applied Economic Analysis, 4005 Locust Av., Long Beach, CA 90807.);

- Numerous Tektronix statistical applications programs for the 4050 series;
- Numerous IBM 5110 resident statistical applications programs; and
- Countless statistical packages and routines on virtually every available turnkey microcomputer system on the market today.

Whether we D^2 and transfer our C^2D&FSs on a micro- or mini-computer system, one configuration would perform like SIR, from Scientific Information Retrieval Inc., a DBMS which interfaces directly with SPSS and BMDP (unfortunately, SIR operates on CDC 6000 and the Cyber series computers). Realistically, however, we cannot expect SIR-like performance from canned mini- and/or microcomputer systems--unless we create it.

4.1.3.4 Display Properties - A quick look at the existing C^2D&FSs indicates that there is no standard output display. Without question, current display properties are utilized because of machine capabilities. By and large font size and differentiation, a useful display characteristic, is not present in our C^2D&FSs simply because our machines do not offer such capabilities. Similarly, because our machines are limited in display type, we make little use of spacing or aspect ratio techniques concentrating instead almost exclusively upon case utilization.

4.1.3.5 Display Coding Techniques - Current C^2D&FSs also seldom utilize several proven techniques for enhancing

man-computer interaction. These techniques include blink coding, motion, depth, and focus (or distortion). Shape and color coding are also only infrequently used. Why? Again we find hardware limitations. But we also find that contractors are relatively unfamiliar with many of these techniques and inexperienced with the programming techniques required for enhanced man-machine relations.

Color coding is a particularly interesting technique because there is no clear evidence that it enhances performance. It is known, however, that color enhances attitudes toward computer usage in the initial stages. Accordingly, since so often at least part of the problem with technology transfer involves "naive" users, color coding may be useful--but should not be expected to contribute to long term performance.

4.1.3.6 Interaction Mode/Dialogue Types - There is no question that full natural language interaction is the best possible interaction mode. Unfortunately, it is some years away. Interactive menu-selection (especially when combined with the use of function keys and a command language) is fine. The use of interactive graphics is also desirable but only when used prudently.

4.2 The Design and Development of C^2 Decision and Forecasting Systems

Thus far we have uncovered a great deal of information relevant to the D^2 of advanced C^2 decision and forecasting systems. Our intention was to examine the whole D^2 process rather than focus only upon display devices, specific software languages, or data base management systems. To the extent that we have achieved this, we have gone far beyond the project's original goals. The time has now arrived, however, to prescribe

the optimal C^2D&FS computer-based configuration. We will present our findings in the same manner and order as we discussed available options and alternatives above. Following this, we will zero in upon specific hardware and software configurations, and present an optimal design and implementation plan for the future.

4.2.1 Requirements Analysis - Requirements analyses must precede the D^2 of all C^2D&FSs. If such analyses cannot be conducted by DARPA personnel then contractors and/or consultants (preferably with real experience) should be hired to do the job. The analyses should be organizational/bureaucratic and substantive. The requirements analytical method should be a hybrid questionnaire/interview/"critical incident technique" method geared to not only uncover requirements, but to determine in no uncertain terms the kind of user likely to actually use the system.

4.2.2 Hardware - We have already determined that for a whole host of reasons the PDP 11/70 is to remain the C^2D&FS's minicomputer. However, real questions arise regarding the necessity of the PDP 11/70. Given the power of today's microcomputer systems, there are very few reasons to use the 11/70 as a transfer system (one can still argue convincingly that there are crunch applications and transfer site requirements reasons for 11/70 use, however--but not for long!).

First, we must decide upon the relative advantages and disadvantages of S-100, non-S-100 and so-called turnkey systems. As suggested in Section 4.1.2.1, S-100 systems are attractive because of the standardized connection. Our research suggests, however, that the S-100 "standard" is now being challenged by the IEEE and GFIB connections. This means that the S-100 "mix and match" capability may soon be equalled via IEEE and GFIB interfaces. Also, even though myriad options are nice, a mixed/

matched system is often less reliable and physically attractive than another (non-S-100 or turnkey) system. Accordingly, our recommendation is to use a S-100 system only when the peripherals are manufactured by the same manufacturer (which ideally also manufactures the processor). Cromemco is such a manufacturer. <u>We explicitly recommend against configuring a system of many diverse parts and manufacturers</u>. While such a system may appear to be ideal on paper, in practice it simply won't work as well as a less complicated system. Finally, a "centipede" system will invite reliability and maintenance problems.

To a much lesser extent these problems exist within the non-S-100 systems, such as the Apple, Challenger, and TRS-80. These systems generally include many "detached" peripherals by the same manufacturer connected via a SS-50, IEEE, or GFIB interface. Non-S-100 systems can be acceptable if they satisfy other requirements, as presented below.

Turnkey systems are the only systems which are <u>integrated</u> in <u>design</u>, <u>function</u> and <u>appearance</u>. They also tend to offer superior compilers and interpreters, and a good deal of applications software (generally ignored by the C^2D&FS community). The IBM 5100, 5110, and 5120, the Tektronix 4050 and 4080 series, and the PERQ family are all turnkey systems.

Perhaps the most important mainframe characteristic is its address space. All of the microcomputer systems popular in the C^2D&FS community are 8 bit processors. <u>There is absolutely no reason whatsoever to continue with these systems when 16 bit microcomputer systems are available</u>.

A final extremely important consideration is storage, both main and mass. The Tektronix and IBM systems, for example, are inferior storage systems. In the future we should select systems which have serious storage capabilities.

4.2.3 Input Devices - The selection of input devices must be dependent upon the results of the requirements analysis and the type of intended user. Intuitively, it would appear that naive users would prefer spatial, touch, or speech input device types. However, our conclusion is that such input devices are not appropriate for "naive" or "commander-type" users because such devices presume interactive computing experience and familiarity which may not exist. Accordingly, keyboard, function key and other related input devices are probably better for naive and quasi-experienced users. This, however, is not to say that programming techniques go unchanged. By and large, C^2D&FS contractors are in a "rut" consisting of "form-filling" and menu-selection approaches. Certainly we can do better. In any case, the use of lightpens, trackballs, mice, and the like should be rejected.

4.2.4 Display Devices - Here too requirements analyses and user profiles should dictate selection. Some general guidelines include the prudent use of graphics and hard copy capability.

Font variation and manipulation should also be possible as should techniques like blinking and coding when--and only when--the requirements analysis and user profile suggest their appropriateness.

4.2.5 Portability - Briefly, portability means realistic transportability. In any case, the target transfer system ought to be microcomputer based; no longer should we "negotiate" for PDP 11 time and space at a transferee's site. It never works and almost complicates irreversibly our D^2 process.

4.2.6 Reliability - Before purchasing a microcomputer system, information should be gathered regarding the system's reliability and--much more importantly, maintenance.

4.2.7 Appearance - Appearance is subjective. However, as suggested above, multi, disconnected pieces generally are relatively unattractive.

4.2.8 Processing Speed - Speed is not a function of the microprocessor but rather overall system configuration--and software.

4.2.9 Software Languages - In our judgement, it is time that the C^2D&FS research community graduated from BASIC. Indeed, advanced degrees are available in APL and especially PASCAL. It is also time that we stopped relying upon language interpretors and began to program and execute via compilers. This judgement is primarily based upon the consideration of execution speed.

4.2.10 Data Base Management Systems - We have spent a good deal of time canvassing the DBMS Systems marketplace. Indeed, we have spent a good deal of time talking with users regarding many systems. Our survey has indicated that there are a number of systems adequate for PDP 11 application.

There are also a number of microcomputer systems which, while degraded from PDP 11 system capabilities, are barely acceptable as data base management systems. None of these systems are as user-oriented as C^2D&FS requirements would demand. Indeed, this is as true for the micro systems as it is for the minicomputers (interestingly, INGRES is not even categorized as a production DBMS). Thus, if our mission is to implement an existing DBMS and expect the interactive hand-holding present in the EWAMS, for example, then we will certainly fail. One simply can not lift a DBMS off the shelf and expect it to be user functional. Training, experience and patience are prerequisites to such use.

If, however, our intention is to hide the DBMS routines within and "behind" an applications program, then the prospects --with significant modifications--are brighter. Indeed, there are only two ways to go: (1) select a mini (and micro) DBMS and rewrite (modify) the routines for C^2D&FS use or (2) identify the key DBM requirements for C^2D&FS applications and write a program which is treated as the C^2D&FS mini and micro "standard." Option 1 may make sense when a solid DBMS exists. For example, SEED and QDMS could be quite easily modified to satisfy C^2D&FS requirements. (However, they would have to be rewritten to some extent to function under UNIX.) On the micro side, both Cromemco and Three Rivers (PERQ) offer DBMSs that are amenable to useful modification, but level of effort estimates can not be made until a thorough analysis of the "innards" of the system can be made. Indeed, it may well be that a new system would be cheaper. Our specific recommendation at this point is to begin with existing DBMSs with a view toward modified system production. If this approach proved imprudent then we could switch to ground zero production.

The C^2D&FS data management requirements are not, by and large, great. Regardless of which option is selected, then, the project is doable. (Requiring contractors to adhere to the new "standard", however, could be difficult...)

4.2.11 Statistical Packages - There are a large number of PDP 11/UNIX and microcomputer statistical "packages." We still feel strongly that SPSS (Version II) and BMDP are the best general purpose statistical packages available for PDP 11 use. In addition, there are some forecasting and simulation packages available for UNIX.

The microcomputer statistical systems are in reality disjointed routines programmed for very few statistical operations. Conversations with users indicate that storage and

execution speed limitations are rampant. Even the Northwest Analytical (NWA) STATPAK is slow and limited in application (see APPENDIX C). In addition, the NWA system requires microsoft BASIC, the CP/M operating system, an 8 inch floppy disc drive, and 48K main memory.

Interestingly, the Tektronix 4050 series turnkey microcomputer systems have statistical routines for numerous operations, but we have not used them for research or developmental (via source code modification) purposes.

We suspect, however, that even if such software systems were much more powerful and user-oriented that they would still not in and of themselves become the bases for C^2D&FS D^2 because the intended product should hide the statistical routines, and not make large input demands upon the user. Not unlike data base management routines, statistical routines should be invisible to the user (unless, of course, the C^2D&FS <u>is</u> a research statistical analytical system). This realization changes our perspective somewhat when we consider the D^2 of generic C^2D&FSs insofar as the real questions have little to so with off-the-shelf statistical systems but with how efficient statistical algorithms (regardless of where they come from) can be used to process data from the DBMS routines. In other words, the task is to isolate those algorithms necessary for statistical processing and then build them into C^2D&FSs, regardless of whether they are adopted from existing routines are written anew.

If we isolate those statistical operations which occur regularly in C^2D&FS execution we see very few. Current systems implicitly or explicitly calculate modes, medians, ranges, means, frequency distributions, some low level correlations, Z-scores, and some covariance analyses. <u>However</u>, the operations

which precede and underlie the systems themselves are much more sophisticated. This is an absolutely critical distinction: on the one hand we have statistical analytical requirements which precede or enhance $C^2D\&FS$ development and/or performance, while on the other we have those operations which occur during on-line system use. Our generic system should be applicable to both of these phases (as should our generic DBMS). Accordingly, we must make a series of hard decisions about which way to go vis-a-vis the two <u>distinct</u> D^2 phases as suggested below:

Mini/Micro Antecent Analyses	Mini/Micro $C^2D\&FS$ On-Line Operations
• DBMS	• DBMS
• Statistical Routines	• Statistical Routines

The distinction further suggests that statistical (and DBMS) requirements during the pre-D^2 phase may be much greater than at the system operation phase where pre-calculated files coupled with unsophisticated statistical operations can essentially constitute "the" system. This in turn suggests that the generic mini- and microcomputer statistical (and DBMS) algorithms intended for on-line execution remain low-level, and that pre-D^2 algorithms be (relatively) powerful.

4.2.12 <u>Display Coding Techniques</u> - Competent requirements analyses should determine the nature and use of display coding techniques such as blinking, motion, depth, and color coding.

4.2.13 <u>Interaction Mode/Dialogue Types</u> - Until full natural language is available "standard" techniques should be used, such as menu-selection via function keys and command languages.

5.0 CONCLUSION

We have already said enough about the need for thorough requirements analyses. The microcomputer mainframe should be a 16 bit one and have ample (at least 5MB mass and 200+KB main) storage. Dual density disks are a prerequisite if floppy; an alternative hard disk is manufactured by Winchester and implemented and marketed by numerous OEMs. As of today, we recommend the PERQ system (acceptable alternatives are the Cromemco System Three and Z2-H). The PERQ is a very high speed 16 bit system (CPU) with integrated I/O controllers. It has 256KB of RAM (with a 1MB RAM option!) and a built-in 12MB rigid disk (with a 24MB option!). By comparison, these capabilities literally dwarf the now too popular Tektronix and IBM systems. Indeed, there is no fair comparison among the systems. The Cromemco (S-100) system is a Z-80 based 8 bit system with up to 512KB of RAM and 2MB of disk storage. The Cromemco Z2-H has 11MB of hard disk and 64KB of RAM (expandable to 512KB). All of these systems may be considered turnkey, if configured consistently.

Input devices are standard with the Cromemco systems, consisting of keyboards and joystick (bad)/function key (good) consoles. The PERQ system provides a (detachable) keyboard and a touch tablet (and speech output).

The Cromemco's display system is grounded primarily in software traits and characteristics (implanted by the programmer not the manufacturer). However, since the Cromemco is an S-100 system there are many display options available. The PERQ graphics display system is a 768 point by 1024 line, bit mapped, raster scanned image on a 15" CRT. All 1024 lines are refreshed 60 times per second for flicker-free high resolution. Font can be any size, shape or complexity (multiple fonts are supported

as well as proportionately spaced characters). In addition, the PERQ uses a display window manager which partitions the screen into separate areas or "windows," which may be moved around the screen, enlarged, or contracted in two dimensions, scrolled, and/or clipped under direct user control. Menus can be inserted into the windows (for continual display during operation) and can be as large as the entire screen or as small as a postage stamp.

Both the Cromemco and PERQ series systems are as portable (if not more so) than the Tektronix and IBM systems (including their disk drives and printers/hard copy units).

Candidly, reliability and maintenance are relatively unknown.

The PERQ is more attractive than the multi-piece Cromemco.

PERQ is extremely fast; Cromemco somewhat slower, but both are much, much faster than the Tektronix or IBM systems.

The Cromemco Z2-H supports extended BASIC, FORTRAN IV, RATFOR, and COBOL in interpreter form. The PERQ has a full PASCAL compiler, thus satisfying our language/speed criterion.

The Cromemco and PERQ systems have DBMSs as part of their software support packages. Our recommendation is to work with these systems to first determine how effective they might be on-line for $C^2D\&FS$ operation. If they prove adequate then no modifications or new programming would be necessary; if not, then the real work would begin.

(On the minicomputer side, we recommend SEED or QDMS for research purposes.)

Statistical operations should be minimized on the PERQ; the PDP 11/70 should support more sophisticated operations in a research mode. In this vein, a bona fide research system, consisting of an integrated DBMS/Statistical System, should be developed coupling SEED/QDMS with BMDP/SPSS(11). This would permit advanced analyses for hypothesis testing and avoid the (re-)writing of individual routines for specific projects.

On the micro side, we should adhere to the same design but not require the system to crunch large data bases or expect the micro to support sophisticated pre-D^2 analyses--unless we are prepared to write new micro "statpacks."

From a design perspective, we recommend a one designated micro per project arrangement. For example, TRAP could be a PERQ-based C^2D&FS and the XAIDS a Cromemco-based system. If projects were "assigned" a hardware/software configuration from the outset then many selection problems could be avoided. This approach would also prevent us from locking ourselves into a particular configuration enabling us to grow and evolve along with the micro market which is expanding rapidly perpetually. PDP 11/70-based systems should no longer be transfer delivery systems; rather, they should be used as research and "inventory" systems. For "delivery" purposes a set of designated micro system DBMS/Statistical Routines should be adopted, modified and/or written for the designated system and standardized.

Potential designated (powerful 8 and 16 bit) systems abound the marketplace and should be very seriously evaluated in the future. We have zeroed in upon PERQ and Cromemco because today they are probably the best systems available for across the board C^2D&FS use. But there may indeed be C^2D&FSs which could be implemented nicely on other systems. Provided these systems made sense from a requirements and hardware/software

standpoints, then they might very well be appropriate (see APPENDIX A).

A final capability undiscussed thusfar is networking. The PERQ's networking capabilities, for example, suggest a new kind of C^2D&FS. User's of one PERQ "station" can access data and programs which run on another (at 10MBits per second via a single coaxial cable using standard cable TV technology). Accordingly, one can imagine a distributed system swapping data and programs to accommodate sharing and shared decision-making and forecasting. Alternatively, a single full blown PERQ station could house several large data sets and supply other PERQ stations with different applications programs. The possible networked configurations are endless.

This journey began with a specific mission: to determine the feasibility of D^2 two integrated, generic DBMS/Statistical Systems. On the minicomputer side we recommend going ahead with the production of a standard system as outlined above. On the micro side we recommend a different approach: designated delivery systems standardized unto themselves and specific projects. We recommend this primarily because of the exploding micro marketplace and the inferiority of existing DBM and statistical systems available for micro use. We also recommend that the PDP 11 be permitted entry into the new C^2D&FS family only as a research system. No longer should CTD export mini-computer-based C^2D&FSs (unless, of course, a "customer" leaves no alternative).

We have also recommended an obvious alternative to the present Tektronix and IBM systems: the PERQ and Cromemco systems. Adoption of the PERQ, for example, gets us out of

the BASIC rut, gives us massive (relative) storage, full, superior graphics (seen chiefly in the window manager and font deviation and manipulation), a PASCAL compiler, and even networking.

All of this is recommended with a view to the future and the design, development and transfer of advanced state-of-the-art--and beyond--C^2 computer-based D&FSs.

6.0 FOOTNOTES

[1]We recognize that the C^2D&FS Program is an evolving one. We are working from one blueprint which may or may not be accurate in every detail; nevertheless, it is accurate in its characterization of the C^2D&FS Program as essentially targetted toward bridging the man-computer gap within C^2 contexts.

[2]See Stephen J. Andriole, "Another Side to C^3", Defense Management Journal, May-June 1979, pp. 15-17.

[3]See Don R. Harris, Albert C. Clarkson, and Gerald Fuller, "The Framework, Process and Functions of C^3I", ESL, Incorporated, Command and Control Project Office, March 20, 1979.

[4]See H. Rudy Ramsey and Michael E. Atwood, "Human Factors in Computer Systems: A Review of the Literature", SAI, Incorporated, September, 1979, pp. 31-32.

[5]Ibid., pp. 31-39.